

Fundamentos de Sistemas de Segurança da Informação

Instrutor: André Sarmiento Barbosa
Coordenador: Luís Felipe M. de Moraes

Atenção: Não é permitida a reprodução desta apostila, ou qualquer parte da mesma, sem o consentimento do autor. Esta apostila é individual e é parte integrante do Treinamento em Sistemas de Segurança da informação.

Aluno: _____

Índice e Planejamento das aulas

1ª aula:

1. Introdução:

- 1.1 Motivação;
- 1.2 Jargão de Segurança;
- 1.3 Necessidade de uma infraestrutura de segurança;
- 1.4 Conceitos Básicos;

2. Oportunidades e Riscos:

- 2.1 Vulnerabilidades;
- 2.2 Comércio Eletrônico;
- 2.3 Conhecendo o inimigo;
- 2.4 Segurança de acesso e de dados;
- 2.5 Itens importantes para a segurança;

2ª aula:

3. As ameaças:

- 3.1 Problemas mais comuns;
- 3.2 *Script Kiddies*;
- 3.3 Engenharia Social;
- 3.4 Vírus, *Worms* e *Trojans*;
- 3.5 Ataques de Negação de Serviço (DoS e DDoS);

4. Metodologia dos ataques:

- 4.1 *Footprinting* – Coletando informações do alvo;
- 4.2 Varredura;
- 4.3 Enumeração;
- 4.4 Ferramentas utilizadas por *hackers*;

4.5 *Exploits e BufferOverflow;*

3ª aula:

5. Contramedidas:

- 5.1 Contramedidas Básicas;
- 5.2 Segurança no nível do cliente;
- 5.3 Configuração segura de serviços;
- 5.4 Resposta a ataques e problemas de segurança;

6. Ferramentas de Segurança:

- 6.1 Controle de acesso;
- 6.2 Criptografia;
- 6.3 Firewalls e IDS;
- 6.4 *Scanners*;
- 6.5 *Sniffers*;
- 6.6 *Password Crackers*;
- 6.7 Logs e Auditoria;

4ª aula:

7. Firewalls:

- 7.1 Tipos de Firewall;
- 7.2 Arquiteturas;
- 7.3 Fabricantes;
- 7.4 Testando a segurança de um firewall (*penetration test*)

8. Sistemas de Detecção de Intrusão (IDS):

- 8.1 Tipos de IDS;
- 8.2 Fabricantes;
- 8.3 Testando o Sistema de detecção de intrusão.

5ª aula:

9. Tecnologias avançadas de segurança:

- 9.1 Secure Sockets Layer (SSL);
- 9.2 IP seguro (IPsec) e Redes Virtuais Privadas (VPNs);
- 9.3 Certificados Digitais;

10. Gerência de Segurança

- 10.1 Plano de Segurança;
- 10.2 Análise de ameaça e Análise de Risco;
- 10.3 Normas de Segurança;

Apêndice A – Glossário de Segurança

1. Introdução:

1.1

utilizações eram como meios de comunicação (usando e-universidades e por empregados em algumas empresas para compartilhar

Mas agora, milhões de corporações, empresas, instituições e usuários domésticos estão usando a Internet para fazer transações bancárias, comércio eletrônico, etc. Portanto, segurança hoje, é um problema potencialmente crítico.

Nós iremos discutir segurança de redes de vários ângulos diferentes, falaremos sobre ameaças, contramedidas, ferramentas, algoritmos, protocolos e gerência de segurança, a fim de tornar os sistemas e a redes mais seguros.

1.2 Jargão de Segurança

Primeiramente, é necessário aprender o jargão utilizado na área de segurança de redes e sistemas. Muitas, senão todas as palavras definidas abaixo serão utilizadas no decorrer deste curso, um glossário pode ser encontrado no Apêndice A.

Ataque:

Evento que pode comprometer a segurança de um sistema ou uma rede. Um ataque pode ter ou não sucesso. Um ataque com sucesso caracteriza uma invasão. Um ataque também pode ser caracterizado por uma ação que tenha um efeito negativo, Ex: DoS.

Autenticação:

É o processo de se confirmar a identidade de um usuário ou um *host*, esta pode ser feita na camada de aplicação (através de uma senha), ou mais complexa, no esquema desafio-resposta (utilizando algoritmos específicos).

Back door:

É um programa escondido, deixado por um intruso, o qual permite futuro acesso à máquina alvo. Este termo é um sinônimo para um termo mais antigo: *trap door*.

Bug

Uma falha, ou fraqueza num programa de computador. Veja *vulnerabilidade*.

Cavalo de Tróia

Uma aplicação ou código que, sem o conhecimento do usuário realiza alguma tarefa que compromete a segurança de um sistema, em geral, esta aplicação se apresenta ao usuário de forma rotineira e legítima, Ex.: um simples jogo que contém código malicioso que envia os dados do usuário para um e-mail específico.

CERT

The Computer Emergency Response Team – Uma organização dedicada a segurança, seu propósito é dar assistência à redes que foram invadidas ou estão sob ataque. Eles podem ser encontrados em <http://www.cert.org>

Certificação

Existem duas definições para este termo. Primeira, certificação pode se referir para o resultado de uma avaliação com sucesso

Crack

Programa utilizado para quebrar licenças de outros programas. Também pode se referir a programas utilizados para quebrar senhas.

Cracker

Indivíduo com conhecimentos elevados de computação e segurança, que os utiliza para fins criminosos, destruição de dados ou interrupção de sistemas. Também pode se referir a programas utilizados para quebrar senhas (Ex. *Password Cracker*).

Engenharia Social

Técnica utilizada por hackers e crackers para obter informações interagindo diretamente com as pessoas.

Exploit

Programas utilizados por hackers e crackers para explorar vulnerabilidades em determinados sistemas, conseguindo assim, acessos com maior privilégio.

Firewall

Equipamento e/ou software utilizado para controlar as conexões (que entram ou saem) de uma rede. Eles podem simplesmente filtrar os pacotes baseados em regras simples, como também fornecer outras funções tais como: NAT, proxy, etc.

Flood

Sobrecarga (em geral, de pacotes) causada por eventos não esperados que causam lentidão da rede.

Hacker

Indivíduo com conhecimentos elevados de computação e segurança, que os utiliza para fins de diversão, interesse, emoção. Em geral, hackers não destroem dados, possuem um código de ética e não buscam ganhos financeiros. O termo hacker é atualmente adotado pela mídia de forma indiscriminada, se referindo a crackers por exemplo.

Hacking

É o ato de hackear sistemas, não no sentido único de invadir, mas principalmente de descobrir como funcionam, e se possuem falhas.

Hijacking

É o assalto de uma sessão, geralmente TCP/IP. O assalto de sessão é uma forma de obter o controle de uma conexão iniciada por um usuário legítimo. Ao interceptar esta conexão o “hacker” pode impedir o usuário legítimo de usar o sistema e tomar o seu lugar.

Hole

Um bug ou uma vulnerabilidade.

Intrusion Detection System -IDS

É um Sistema de Detecção de Intrusão, um software responsável por monitorar uma rede ou sistema e alertar sobre possíveis invasões.

Invasão

Caracteriza um ataque bem sucedido.

Lammer

É uma palavra que os hackers utilizam para identificar os indivíduos que se acham hackers, mas estão ainda no estágio inicial de aprendizado.

Phreaking

São os hackers de telefonia, convencional ou celular.

Scanner

Ferramenta utilizada por hackers ou especialistas em segurança que serve para “varrer” uma máquina ou uma rede, em busca de portas abertas, informações ou serviços vulneráveis.

Script Kiddie

É o indivíduo que saiu do estágio de *lammer* mas que só sabe usar as “receitas de bolo”, programas prontos e ainda não entende muito bem o que está fazendo.

Sniffer

Ferramenta utilizada por hackers e especialistas em segurança e de rede que serve para monitorar e gravar pacotes que trafegam pela rede, dependendo do sniffer, é possível analisar vários dados dos pacotes, analisar protocolos, ver dados específicos da camada de aplicação, senhas, etc.

Spoofing

É uma forma de manter uma conexão com uma máquina se fazendo passar por outra na qual ela confie. Um termo muito utilizado é o *IP Spoofing*, que significa o uso de vulnerabilidades do Protocolo TCP/IP que permitem a ação descrita acima.

Trojan, Trojan Horse

São os cavalos de tróia, programas que são entregues para o usuário de forma legítima (muitas vezes podem ser coisas interessantes como joguinhos, cartões virtuais, etc.), mas que internamente realizam ações maliciosas, tais como: gravar senhas, gravar toques de tecla, e posteriormente armazenar estas informações ou enviar para outra pessoa.

Vírus

São códigos ou programas que infectam outros programas e se multiplicam, na maioria das vezes podem causar danos aos sistemas infectados.

Vulnerabilidade

Estado de um componente de um sistema que compromete a segurança de todo o sistema, uma vulnerabilidade existe sempre, até que seja corrigida, existem vulnerabilidades que são intrínsecas ao sistema. Um ataque explora uma vulnerabilidade.

Warez

Nome utilizado por hackers para se referir a pirataria de software.

Worm

Um worm é semelhante a um vírus, mas difere pelo fato de não necessitar de um programa específico para se infectar e reproduzir. Muitos vírus hoje, possuem a característica de worms e vice e versa.

1.3 Necessidade de uma infraestrutura de segurança

As vulnerabilidades existem, os ataques também existem e crescem a cada dia, tanto em quantidade quanto em qualidade. Uma infraestrutura de segurança não é só necessária como obrigatória, devendo existir, além de um investimento específico, um planejamento, uma gerência e uma metodologia bem definida.

É importante lembrar que os seres humanos são o elo mais fraco dessa corrente. Você pode ter os mais perfeitos produtos de segurança, mas eles não serão nada se você não tiver consciência que eles serão gerenciados e utilizados por pessoas, isto nos faz refletir sobre a necessidade de uma infraestrutura de segurança e forma de como montá-la.

A modelagem de ameaça é a primeira etapa em qualquer solução de segurança. Quais são as ameaças reais contra o sistema? Se você não souber disso, como saberá quais tipos de contramedidas irá empregar?

A modelagem de ameaça é algo difícil de ser feito, ela envolve pensar sobre um sistema e imaginar o vasto panorama de vulnerabilidade.

Avaliação de Risco:

Não é suficiente apenas listar um punhado de ameaças, você precisa saber o quanto deve se preocupar com cada uma delas.

Ao projetar um sistema de segurança, é vital fazer esse tipo de modelagem de ameaça e avaliação de risco. Muitos projetistas de sistemas pensam no projeto de segurança como um livro de receitas: misture contramedidas específicas – criptografia e *firewalls* são bons exemplos – e magicamente você estará seguro, isso nunca acontece!

Não entraremos em detalhes nesta etapa inicial, vamos começar conhecendo as vulnerabilidades e depois nossos inimigos, então conheceremos as ferramentas de defesa e as tecnologias, depois disso sim, teremos uma noção bem melhor de como devemos nos proteger e elaborar um projeto de segurança.

1.4 Conceitos Básicos

Nesta seção abordaremos alguns conceitos que não são simplesmente definições de termos, são pilares de um sistema seguro, sendo que alguns estão intimamente ligados ao estudo da criptografia, são eles:

Privacidade Integridade Autenticação Não-repúdio Disponibilidade
--

A criptografia é a arte de guardar mensagens seguras, ou transmiti-las de forma segura, isto garante a *privacidade*. A criptografia também pode trazer outros benefícios à segurança, tais como:

- Integridade: Deve ser possível ao receptor de uma mensagem verificar se esta foi alterada durante o trânsito.
- Autenticação: Deve ser possível ao receptor de uma mensagem, verificar corretamente sua origem, um intruso não pode se fazer passar (personificar) o remetente desta mensagem.
- Não-repúdio ou irrevogabilidade: O remetente de uma mensagem não deve ser capaz de negar que enviou a mensagem.

A disponibilidade se refere ao sistema estar sempre pronto a responder requisições de usuários legítimos, isto envolve o uso de sistemas redundantes, no-breaks e mecanismos contra ataques de *Denial of Service*.

2. Oportunidades e Riscos:

2.1 Vulnerabilidades

Oportunidades são o que não faltam na vasta gama de serviços disponíveis hoje na Internet. A começar pelo número de usuários que temos na Internet, o qual é praticamente impossível de se mensurar, e ainda pelo fato dela ser a maior compradora e vendedora do mundo. Ela também tem o potencial para ser a maior projetista, a maior desenvolvedora, a maior fabricante e a maior distribuidora.

Vamos iniciar citando algumas vulnerabilidades que são semelhantes ao mundo não-virtual (fora da Internet): Pornografia, jogos de azar, falta de privacidade e assédio sexual, são apenas algumas. As ameaças específicas da Internet seriam os vírus, worms, trojans e o *hacker* de computador.

Sem dúvida o tipo de ameaça mais comentada é o hacker de computador. Não só pelo caráter interessante do fato, mas também pela quantidade de problemas que um hacker mal intencionado (*cracker*) pode causar.

Dentre as coisas terríveis que podem acontecer podemos apontar:

- Suas senhas poderão ser roubadas e sua conta modificada;
- Suas linhas de comunicação podem ser grampeadas e informações secretas de sua empresa poderão ser comprometidas;
- Sua rede poderá ser inundada de informações e entrar em colapso;
- Sua propriedade intelectual poderá se roubada;
- Você poderá estar violando controles de exportação de tecnologia;
- Seus funcionários poderão ser pegos transportando pornografia;
- Suas finanças poderão ser alteradas;
- Pode haver falsificação de dinheiro;
- Alguém poderá se passar por você e efetuar transações financeiras em seu nome;

Os pontos vulneráveis descritos anteriormente decorrem de fatores mais técnicos que dependendo do caso podem ser extremamente complexos. Os hackers conhecem estes fatores e sua complexidade, nós, que desejamos proteger nossas redes também temos que conhecer tanto quanto eles, e por isso, temos que entender as vulnerabilidades do ponto de vista técnico.

Quando uma vulnerabilidade é divulgada, isto é, alguém a descobriu e a divulgou, o descobridor irá pertencer a um destes três grupos:

- Hackers
- Crackers
- Fabricantes de software ou hardware

Como estar informado sobre tecnicamente sobre as vulnerabilidades?

Esta é uma pergunta muito fácil: Pela própria Internet. Assine listas de discussão específicas e visite *sites* de segurança todos o dias,

Algumas Listas de discussão:

BUGTRAQ (pertence a SecurityFocus).

BOS-BR (Brasileira, pertence a Securenet)

Alguns Sites:

<http://www.securityfocus.com>

<http://packetstorm.securify.com>

<http://ciac.llnl.gov>

<http://www.lockabit.coppe.ufrj.br>

<http://www.securenet.com.br>

Uma lista bem grande de sites de segurança pode ser encontrada na seção de links do portal Lockabit da COPPE/UFRJ.

É importante verificar também continuamente os sites dos fabricantes dos softwares que sua empresa utiliza, normalmente eles possuem uma seção dedicada a segurança.

2.2 Comércio Eletrônico

O comércio eletrônico, a compra e venda de mercadorias e serviços através de um meio eletrônico, esta será uma das mais importantes maneiras de se fazer negócio no futuro. Nos últimos tempos, porém, o progresso tem sido mais lento devido a falta de mecanismos seguros para efetuar pagamentos eletronicamente na Internet.

Mesmo com tecnologias já amadurecidas, ainda temos problemas com o fator humano que nega a confiar em tais tecnologias. As pessoas acham que estão mais seguras

usando o cartão de crédito numa loja onde o atendente leva seu cartão e volta minutos depois do que na Internet. Às vezes essas pessoas estão erradas.

Como pesquisar soluções para comércio eletrônico.

Enquanto estiver analisando protocolos e tecnologias de pagamento, é essencial manter-se atualizado, recorra à própria Internet para obter as informações mais recentes. Também recomendamos que sua empresa ou organização faça parte de grupos de trabalho para padronização de tecnologias. Se a participação direta não for possível, pelo menos mantenha-se a par dos novos desenvolvimentos. A maioria destes grupos possui sites e/ou grupos de discussão dos quais você pode participar.

Ex:

- IETF (Internet Engineering Task Force)
- Comitês ANSI do NIST
- W3C (World Wide Web Consortium)
- CommerceNet
- FSTC (Financial Services Technology Consortium)
- Smart Card Forum

2.3 Conhecendo o inimigo

Muitos problemas de segurança são causados por pessoas maliciosas tentando ganhar algum tipo de benefício, seja ele monetário, político, intelectual, etc.

Vamos considerar a seguinte tabela que mostra alguns adversários que um profissional em segurança terá que enfrentar:

Adversários	Objetivos
Estudante	Divertir-se vasculhando emails, roubando senhas, etc.
Hacker Cracker	Testar a segurança de sistemas, roubar informações
Homens de negócios	Descobrir planos e estratégias de marketing do concorrente
Ex-empregado	Vingar-se por ter sido despedido
Espião	Conhecer a tecnologia do inimigo
Terrorista	Roubar e utilizar segredos políticos

Tabela 1 – adversários, atacantes.

Outro problema grave é a quantidade de informações existente na Internet hoje abordando falhas de segurança e técnicas de invasão. Muitos manuais e ferramentas estão disponíveis, distribuídas por grupos de *hackers* e até por organizações dedicadas a segurança.

Alguns sites de grupos de hackers:

<http://www.ussrback.org>
<http://www.destroy.net>
<http://www.insecure.org>
<http://www.hackershomepage.com>
<http://www.hackerslab.com>
<http://packetstorm.securify.com>
<http://www.rootshell.com>
<http://www.securityportal.com>
<http://www.technotronic.com>
<http://unsekurity.virtualave.net>
<http://www.phrack.com>
<http://www.bufferoverflow.org>

2.4 Segurança de acesso e de dados

A distinção entre estes dois tipos de segurança é muito importante. Temos que ter em mente que um hacker mal intencionado irá descobrir falhas ou usar de qualquer artifício para conseguir quebrar um ou o outro tipo, e ele quebrar uma das duas, conseqüentemente quebrará as duas!

É necessário “trancar” bem a periferia de sua rede ou sistema, mas para fazer você tem que conhece la bem, saber que tipos de dados trafegam e onde você terá que apl

- Que você nunca estará 100% seguro (e isso é verdade)
- Em aprender a pensar como um hacker
- Em usar criptografia e
- Em usar senhas fortes e jamais informar uma senha ou detalhes do seu sistema
- Desconfiar de tudo (e de todos)
-

2.5

aquele que tem as melhores e mais caras ferramentas de segurança.

As ferramentas estão entre um dos itens importantes, mas estão longe de serem as únicas necessárias.

Itens importantes para a segurança:

- Investimento
- Plano de Segurança
- Pessoal qualificado
- Interação na organização e entre fornecedor e cliente
- Ferramentas de segurança

3. As ameaças:

3.1 Problemas mais comuns

As raízes dos problemas de segurança:

- Má configuração dos *hosts*;
- Falhas inerentes dos sistemas;
- Deficiência na reposta dos fabricantes;
- Pobre educação em segurança;

3.1.1 Má configuração dos hosts

A má configuração pode ocorrer devido aos problemas em especial:

- o Configuração *default* do sistema deixa muito a desejar em segurança;
- o Instalação de um sistema sem um mínimo de preocupação com a instalação default: Ex.: Daemons do UNIX, Registros do NT.
- o Instalação e/ou habilitação de serviços de forma indiscriminada

3.1.2 Falhas inerentes dos sistemas

Nestes casos, a culpa é colocada sempre nos fabricantes, porquê seus sistemas possuem vulnerabilidade e falhas, quando não deveriam ter. Bem, acontece que bugs são tão naturais em softwares quanto doenças são em nós, seres humanos. Os bugs, falhas, e vulnerabilidades sempre irão existir, então, cabe a nós, nos manter atualizados quanto ao lançamento de correções, patches, updates etc.

3.1.3 Deficiência na reposta dos fabricantes

Este sim, é um problema causado pelo fabricante, quando este deixa de manter um controle de qualidade de seu software, e o que é pior, não alerta os usuários e nem lança correções para tais problemas. Portanto, antes de adquirir um software, verifique, além de sua funcionalidade, se existe uma área de suporte ativa que cuide das atualizações e tenha preocupação com a segurança do produto.

3.1.4 Pobre educação em segurança

É um dos primeiros problemas que devem ser atacados na implementação de um plano de segurança. De nada adianta termos os melhores profissionais na administração, os melhores produtos, se nossos funcionários não estão cientes da real necessidade de segurança, e como se deve proceder. Não que seja necessário deixar

para não cometer erros que possam comprometer a segurança de todo o sistema.

3.2 Script Kiddies

fácil. Eles não procuram por maneira mais fácil possível. Eles fazem isto utilizando um pequeno número de , então eles procuram pela Internet inteira, até que conseguem máquina que seja vulnerável (cedo ou tarde isto certamente acontecerá).

deixam para trás as sofisticadas. Alguns, não tem a mínima idéia do que o fazendo. Embora o nível técnico deles possa ser diferente, todos eles usam uma posteriormente eles possam explorar estas falhas.

uma ameaça. Cedo ou tarde evitar isto.

seu favor. Com milhões de sistema poderia encontrar você facilmente. Mas este não é o caso. Muitas das ferramentas são usá las. O rápido crescimento do número de ferramentas é alarmante. Como a Internet não tem fronteiras geográficas, esta números está se voltando contra nós. Com o número crescente de usuários rede, não mais uma questão de "Se" mas sim de "Quando" você será testado ou

Este é um excelente exemplo de como a "segurança por obscuridade" pode falhar.

escaneá lo?. São estes sistemas que os estão procurando, pois um

Os hackers mais eles invadem o sistema. As Backdoors permitem acesso fácil e "furtivo" ao sistema, não apareceria em qualquer ele constrói uma "casa" confortável e segura, onde ele pode escanear a Internet

Estes ataques não são limitados a uma certa hora do dia. Muitos administradores os seus logs por testes ou que possam ter ocorrido tarde da

a qualquer hora do dia. Eles também podem ficar escaneando 24 horas por dia. Você não pode ter idéia de quando o *scan* ocorrerá. Pois estes ataques são lançados de qualquer lugar do mundo, e, assim como a Internet não tem fronteiras geográficas, também não existe um tempo fixo. O hacker pode lançar um ataque a meia noite no país dele, mas para você, aqui no Brasil podem ser 3 horas da tarde.

Uma maneira de você se proteger é executar somente os serviços que são necessários. Se você não precisa de um serviço, desligue-o. Se você precisa do serviço, então verifique se você possui a última versão ou se você tem todos os patches/fixes instalados.

Os servidores DNS são muito usados para construir bases de dados dos sistemas que podem ser testados/escaneados. Limite os sistemas que podem fazer a transferência de zona dos seus Servidores de DNS. É altamente recomendado atualizar para a última versão do BIND, que você pode encontrar em <http://www.isc.org/bind.html>. E por último, observe se os seus sistemas sofrem ataques de *scanning*.

O script kiddie é uma ameaça a todos os sistemas, eles não têm nenhum preconceito, escaneiam qualquer sistema, em qualquer lugar do mundo, independente do valor do sistema. Cedo ou tarde seu sistema será testado/escaneado.

3.3 Engenharia Social

Existe algum método mais rápido e eficiente de se descobrir uma senha? Que tal simplesmente perguntar? Por mais extraordinário que possa parecer, o método mais simples, mais usado e talvez mais eficiente de se recolher informações é simplesmente chegar e perguntar.

Você também poderia subornar, mas dependendo da situação, isto pode lhe custar muito caro, então porquê não tentar enganar e obter tais informações? De fato, este método é bastante utilizado, e existem *hackers* que sabem usá-lo com grande destreza, um exemplo é o famoso hacker Kevin Mitnick que era um expert em fazer tais “vigarices”.

Essa tática de ataque é conhecida como “Engenharia Social”. Basicamente, esta é a arte de fazer com que outras pessoas concordem com você e atendam aos seus pedidos ou desejos, mesmo que você não tenha autoridade para tal. Popularmente, pode-se dizer que engenharia social é simplesmente a arte de se contar uma mentira bastante convincente.

Dentro da área de segurança podemos definir engenharia social como a aquisição de informações preciosas ou privilégios de acesso por “alguém de fora”, baseado em uma relação de confiança estabelecida, inapropriadamente, com “alguém de dentro”. Profissionais utilizam este tipo de aproximação para adquirir informações confidenciais, como organogramas de organizações, números de cartões de crédito e telefone, senhas de acesso, diagrama da rede, etc. com o objetivo de avaliar as vulnerabilidades de uma organização para futuros ataques.

Geralmente este tipo de aproximação envolve muito mais do que simples raciocínio rápido e uma variedade de frases na ponta da língua. Engenharia social pode envolver muito trabalho de aquisição de informação antes de uma real ação de qualquer tipo. Para se iniciar um ataque, a maior parte do trabalho está na preparação, muito mais que no próprio ataque.

Dizem que o único computador totalmente seguro é aquele desligado da tomada. A arte da engenharia social concentra-se no elo mais fraco da corrente da segurança de computadores: os seres humanos. O simples fato de que se pode facilmente convencer uma pessoa a ligar o computador, torna vulnerável, até mesmo, os computadores desligados.

Na medida em que a parte humana de um sistema de segurança é a mais essencial, não existe computador na face da Terra que não necessite de seres humanos. Isso significa que essa é uma fraqueza universal, independente de plataforma, software, tipo de conexão de rede ou idade do equipamento. Qualquer pessoa com acesso à qualquer parte do sistema, física ou remota, pode ser uma falha de segurança em potencial. Qualquer informação adquirida pode ser utilizada para um outro ataque de engenharia social. Isso significa que qualquer pessoa, mesmo que não seja considerada integrante da política de segurança pode servir como uma porta de entrada.

O primeiro método é também o mais óbvio. Um pedido simples e direto, onde se solicita ao indivíduo alvo que se execute uma determinada tarefa. Embora este método seja o menos provável a trazer um resultado positivo, é com certeza o mais simples, onde o indivíduo sabe exatamente o que você quer que ele faça.

O segundo é criar uma situação onde o indivíduo é apenas uma parte dela. Com muito mais fatores que um simples pedido, o indivíduo preocupado estará bem mais predisposto a ser persuadido. Isso não significa que as situações propostas devam ser fictícias. Quanto menos você faltar com a verdade melhor. Isso requer muito mais trabalho por parte de quem faz o ataque e com certeza envolve um recolhimento de informação e conhecimento prévio do alvo.

Se a situação proposta, real ou imaginária possuir certas características, o indivíduo alvo estará mais propenso a concordar com o seu pedido. Estas características incluem:

- Difusão da responsabilidade. Fazer com que o alvo acredite que ele não é o único responsável por suas ações e pelas informações que ele possa divulgar. Mantenha a responsabilidade longe do alvo.
- Troca de favores. Permitir que o alvo acredite que esta prestando um favor a você e que você é extremamente grato. As pessoas geralmente mostram-se mais dispostas a cooperar quando acreditam que poderão obter alguma vantagem no futuro, como se você ou o chefe ficassem devendo um favor.

- Dever moral. É quando o alvo coopera, pois acha que é a coisa certa a fazer. É seu dever moral. Parte disso é culpa. As pessoas procuram evitar o sentimento de culpa e farão o possível para evitar esse sentimento.

Procure escolher seu alvo levando em consideração seu envolvimento (veremos mais à frente no texto), sua experiência e tempo de trabalho junto ao sistema alvo. Alunos, estagiários, secretarias e profissionais iniciantes mostram-se sempre mais dispostos a cooperar. Isto se deve ao fato de que estes indivíduos possuem ainda pouco conhecimento e pouca experiência a respeito do sistema alvo e desejam mostrar-se úteis. Eles querem “mostrar serviço”.

Quanto menos conflito com o alvo melhor. É muito mais fácil ganhar a confiança do alvo sendo gentil. Utilizar um tom de voz calmo (se ao telefone) e ser gentil, é um bom começo para que o alvo coopere.

Como uma ataque de engenharia social pode revelar muitas informações, como se pode tornar um sistema de computadores mais seguro? A resposta é educação e difusão da informação, explicando aos empregados e pessoas ligadas direta ou indiretamente ao sistema a importância de uma política de segurança, evitando assim o ataque de pessoas que poderão tentar manipulá-los para ganhar acesso a informações privadas. Isto já é um excelente começo para tornar segura sua rede ou sistema.

3.4 Vírus, Worms e Trojans

Todos os anos, os vírus causam muitos prejuízos ao redor do mundo. A internet é o meio ideal para transmissão destas "pestes virtuais". Na década de oitenta, por exemplo, para conseguirmos transmitir um vírus tínhamos poucos recursos. O mais usado era transmiti-los de máquina em máquina através de disquetes, o que tornava a contaminação bastante lenta. Atualmente as coisas são bem diferentes, através da internet a contaminação é muito mais rápida e atinge facilmente nível mundial. Além disso, surgiram conceitos novos como vírus de macro, worms e trojans.

Estaremos abordando os vírus e suas variantes no ambiente Windows. Existem vírus nos ambientes UNIX, mas a proporção é infinitamente menor. Além disso, os conceitos abordados aqui se aplicam em ambos os casos.

A seguir, um pequeno esclarecimento sobre as diferenças entre os vários invasores que podem vir a nos incomodar:

- VÍRUS - São pequenos programas que, como os vírus biológicos, têm a propriedade de se juntar a outros arquivos, alterar seu funcionamento normal e reproduzir (fazer cópias de si), contaminando outros arquivos. Em princípio um vírus poderia contaminar qualquer arquivo. No entanto, só faz sentido contaminar arquivos executáveis, uma vez que estes são carregados e executados na memória. Por exemplo, vamos supor que um vírus contamine o COMMAND.COM, um arquivo executável que é carregado pelo computador toda vez que nós ligamos o micro. Desta forma, o programador que fez o vírus sabe

que sua "criatura" sempre vai ser carregada na memória. Já se fosse um arquivo de dados como, por exemplo, MYFILE.TXT, nada aconteceria, pois o micro em princípio não executa arquivos com extensão TXT.

- TROJAN - São Cavalos de Tróia, assim como na história, envia-se um falso presente para a vítima (geralmente por e-mail), que ingenuamente aceita e o executa. Assim o Trojan começa a fazer seu ataque se enviando por e-mail para outras pessoas de sua lista, como se fosse o usuário (esta está sendo um pratica cada vez mais comum também por parte dos vírus). Mas há uma diferença fundamental entre os Trojans e os Vírus: os primeiros não se reproduzem como os vírus.
- VÍRUS DE MACRO - Primeiro vamos esclarecer o que é uma macro: Macro é uma VBA - Visual Basic Application (linguagem script desenvolvida pela Microsoft), que é interpretada pelo MS OFFICE (Word, Excel, Access, Power Point), ou seja, podemos fazer pequenos programas que nos ajudem a trabalhar no Office. Como por exemplo, criar um botão na barra de ferramentas do Word que permita abrir diretamente a calculadora do Windows. No entanto, nas mãos erradas, se torna uma arma poderosa capaz de causar muitos prejuízos. Agora é muito fácil entender que um Vírus de Macro nada mais é que um programa escrito em VBA. No momento que abrimos um documento do Word contaminado, esta macro é ativada, podendo apagar documentos importantes, por exemplo.
- WORMS - Os worms são programas que aproveitam falhas do sistema para se propagar, e se replicar. Ao contrário dos Trojans, os Worms não contaminam arquivos. O primeiro Worm que se tem notícia foi criado por Robert Morris, em 1988. Este programa aproveitava uma falha do finger daemon do UNIX e do sendmail. Mas o worm de Morris tinha um bug que o fazia reinfectar máquinas já contaminadas. Isso provocou a queda de vários computadores em varias instituições nos Estados Unidos.

Atualmente está cada vez mais difícil classificar um programa malicioso em uma destas categorias, pois os "vírus" modernos estão usando cada vez mais técnicas mistas de contaminação. Não é raro encontrar programas que usam técnicas de worms para entrar no sistema, alterar as configurações de segurança e infectar seu computador como se fosse um vírus de macro. Está se tornando cada vez mais comum encontrar vírus que usam arquivos de lote (.BAT) para se infiltrar no sistema.

3.4.1 Antivírus

Os vírus acabaram por formar uma grande indústria de antivírus. Atualmente existem da McAfee.

Todos os antivírus agem de forma semelhante. Existem dois métodos básicos usados para combater vírus. O primeiro consiste em manter nos antivírus de dados onde ficam registradas todas as assinaturas (parte do vírus que o caracteriza) de vírus conhecidos. Daí a importância de manter seu antivírus

atualizado, pois a cada dia surgem centenas de novos vírus. Assim, quando *scaneamos* o sistema, na verdade o que estamos fazendo é comparar cada arquivo nosso com a assinatura dos vírus registrados.

A segunda forma de proteção é conhecida como inoculação, que nada mais é que a criação de um banco de dados contendo as principais informações (tamanho, data de criação e data da última alteração) sobre os arquivos inoculados. Assim, cada vez que *scaneamos* o sistema o programa antivírus compara as informações do banco de dados criado com as que estão no disco. Se houver alguma diferença é emitido um alerta. Mas note que não é qualquer arquivo que deve ser inoculado, uma vez que arquivos de dados sempre são alterados. Os arquivos executáveis, DLLs e arquivos de sistema são exemplos de arquivos que devem ser inoculados, pois são as principais vítimas de vírus e não mudam seu conteúdo com frequência.

3.4.2 Algumas Medidas de Segurança

Nenhuma empresa está livre de ataques de vírus. Mas existem algumas medidas que ao menos podem diminuir o risco de contaminação. Em primeiro lugar é muito importante que haja uma conscientização por parte dos funcionários sobre as normas de segurança. Este é o primeiro passo para evitar problemas futuros.

Nada adianta uma equipe super treinada em segurança se os funcionários insistirem em baixar arquivos de origem duvidosa na Rede externa, ou inserirem discos inseguros nos micros. Um dos pontos mais importante do processo de conscientização dos funcionários é a questão do e-mail, pois este é o principal “trampolim” dos vírus atualmente.

Algumas medidas simples podem evitar muita dor de cabeça futura, tais como:

- Não abrir e-mails de pessoas desconhecidas.
- Não abrir arquivos executáveis anexados a e-mails, mesmo que venham de pessoas conhecidas.
- Não abrir documentos do Office contendo macros, se abrir, desabilitar as macros.
- Não baixar programas de sites da internet.
- Não usar disquetes de dentro da empresa em computadores de segurança duvidosa.

Apesar de tudo, o ideal é também ter uma equipe preparada para agir em caso de contaminação. Esta equipe deve se manter atualizada e não só tratar de contaminações, mas também da segurança do site em geral. Algumas atribuições básicas de uma equipe de segurança são:

- Manter o antivírus sempre atualizado.
- Fazer *backups* periódicos.
- Configurar os clientes de e-mail para não interpretarem HTML ou qualquer script.
- Configurar o Office para não executar macros sem permissão.
- Atualizar o Office periodicamente (cada dia que passa descobrem-se novas falhas, que podem ser exploradas).

3.5

cação na internet é baseada no fluxo de pacotes de dados. Mas o que acontece se uma máquina receber mais dados do que ela consegue lidar? ela possui uma quantidade enorme de informação para manipular e portanto, ficará -se o nome de DoS (Denial of Service, ou

após o “flood” recebidos, ocorre a negação de serviço.

O DoS tem sido usado por profissionais de segurança como ferramenta para avaliar a capacidade de sua rede. Por outro lado, do mundo todo têm trazido muitos problemas a pequenos e até grandes sites, como Yahoo redes.

O poder de sobrecarga desses ataques aumenta quando eles vêm de várias máquinas para um alvo, ou seja, o envio de pacotes parte de vários pontos. Trata-se do Distributed DoS (DDoS).

3.5.1 Como funciona o DDoS:

A idéia é instalar programas para ataque DoS em diferentes hosts. Estes computador central, esses Zumbis começam a enviar o máximo de pacotes ao alvo.

Esta idéia pode ser expandida no sentido de sendo que na primeira está o *cracker* na última camada as máquinas às quais irão enviar diretamente os pacotes que irão causar o na vítima.

Mas antes de existirem ferramentas que automatizassem esse ataque, era necessário usar para disparar o ataque. Usando telnet ou SSH, o hacker dispararia o comando para causar o na máquina vítima. Para isso, ele poderia usar, por exemplo, o *ping* do UNIX nos hosts.

3.5.2 Características do DDoS:

Não satisfeitos apenas com o *flood* para apagar os rastros do ataque e tornar o ataque ainda mais poderoso. Para deficiência no protocolo da internet.

3.4.3

Vulnerabilidades do TCP/IP são a chave para o desenvolvimento de novos programas, cada vez mais poderosos nesses ataques. Uma delas, é o Stream Attack, descoberta por Tim Yardley. Esta categoria de ataque funciona da seguinte maneira: na conexão, pacotes são enviados com os indicadores (*flags*) ACK e SYN ligados ou apenas o ACK ligado. Devido a estes *flags* não fazerem parte de uma primeira etapa da conexão, a máquina alvo ficará confusa e levará algum tempo para processar tais dados. Imagine então o *Stream Attack* partindo de várias hosts (modo distribuído), isto ocasionaria uma sobrecarga utilizando-se um menor número de hosts (Zumbis) que o DDoS “padrão”.

3.5.4 Tipos de Ataques de DoS:

- Consumo de Largura de Banda - Neste tipo de ataque, existem pelo menos duas possibilidades:
 - o O atacante possui uma largura de banda maior que a da vítima, o que facilita a inundação.
 - o O atacante agrega outros computadores para que juntos, disparem o ataque, amplificando seu efeito e consumindo a largura de banda do computador vítima.
- Consumo dos Recursos - A idéia aqui é esgotar os recursos do sistema, tais como memória, cpu, quotas de disco, etc., o que pode ocasionar travamento dos processos, entupimento de discos, indisponibilização de recursos.
- Ataques a Servidores de Nomes de Domínios (DNS) e a Roteadores - No caso do ataque baseado em roteamento, o atacante manipula a tabela de roteamento com a finalidade de negar serviço a quem consultá-la, explorando falhas dos protocolos de roteamento, como o Protocolo de Informações de Roteamento (RIP) e o Protocolo de Gateway de Fronteira (BGP). Com isso, o atacante pode direcionar todo tráfego para a máquina dele, ou mesmo para uma rede que não existe, o que chamamos de buraco negro. Assim como nos ataques baseados em roteamento, o ataque a DNS permite direcionar o tráfego. Porém, esses ataques, em sua maioria, consistem em armazenar endereços falsos no *cache* do servidor da vítima.

3.5.5 Exemplos de Ataques DoS:

- SMURF - O ataque *Smurf* é um dos mais temidos. Envolvendo vítima, atacante e uma rede auxiliar, funciona da seguinte maneira: são enviados pacotes ICMP *echo* a rede auxiliar. Porém, a origem desses pacotes é falsificada como sendo o endereço da vítima e quando os pacotes chegam a rede auxiliar, eles serão multiplicados e, portanto, a vítima será inundada com quantos pacotes forem ecoados na rede.
- SYN FLOOD - Para entendermos este ataque precisamos ver como funciona uma conexão TCP entre duas máquinas A e B, que é realizada em 3 etapas. Primeiramente, a máquina A envia um pacote SYN. A máquina B então responde com um outro pacote SYN/ACK que ao chegar a máquina A, reenvia um pacote ACK e então a conexão é estabelecida. A vulnerabilidade

explorada é que a maioria dos sistemas aloca uma quantidade finita de recursos para cada conexão potencial. Mesmo que um servidor seja capaz de atender muitas conexões concorrentes para uma porta específica (Ex. porta 80), o que o atacante explora é que apenas cerca de 10 conexões potenciais são tratáveis. Iniciando o ataque, o *cracker* envia um pacote SYN com origem falsificada (buraco negro), o que deixará a vítima procurando por algum tempo (que varia de acordo com o sistema) a origem para enviar o pacote SYN/ACK. Sendo assim, esta possível conexão fica alocada na fila, que é bastante limitada.

3.5.6 Detectando e evitando:

Há dois tipos de tráfego gerado por DDoS: tráfego de controle (entre cliente e servidor) e tráfego “*flood*” (entre servidor DDoS e a vítima). Para habilitar uma eficiente detecção deve-se procurar por sinais gerais (assinaturas), alguns óbvios, outros pelo volume de tráfego e que causam suspeita.

Ainda não existe uma solução para bloquear um ataque DoS/DDoS. O que se pode fazer é tentar minimizar seu impacto, para fazer isso temos que primeiro identificar corretamente um ataque de DoS e depois criar soluções para “escoar” o fluxo de pacotes, seja através de um firewall na fronteira ou algum esquema de alteração de endereçamento IP ou DNS.

4. Metodologia dos ataques:

4.1 *Footprinting* – Coletando informações do alvo

Antes da diversão real do Hacker começar, três passos especiais, e às vezes árduos, precisam ser executados. Falaremos agora sobre o primeiro: a aquisição de alvos, ou *footprinting*, a fina arte de coletar informações do alvo. Por exemplo, quando um ladrão decide roubar um banco ele não entra simplesmente no banco e começa a exigir o dinheiro (pelo menos, não os expertos). Em vez disso, eles fazem grandes esforços para obter informações sobre o banco – as rotas dos carros-fortes e horários de entrega de dinheiro, as câmeras de vídeo, o número de caixas, etc.

O mesmo se aplica a invasores bem-sucedidos. Eles precisam colher uma grande quantidade de informações para executar um ataque cirúrgico e bem direcionado (um que não seja descoberto imediatamente). Como resultado, um atacante irá coletar o máximo de informações sobre todos os aspectos de segurança de uma organização. Eles acabarão obtendo um *footprint* (perfil) único de sua presença na Internet. Seguindo uma metodologia estruturada, atacantes podem juntar sistematicamente informações de uma grande variedade de fontes e compilar esse *footprint* crítico de qualquer organização.

Embora haja diferentes técnicas de *footprinting*, seu objetivo primário é descobrir informações relacionadas a tecnologias de Internet, intranet, acesso remoto e extranet. A tabela a seguir mostra essas tecnologias e informações críticas que um atacante tentará identificar:

Tecnologia	
Internet	Nome de domínio; Blocos de rede; Endereços IP; Serviços TCP e UDP executados em cada sistema identificado; Arquitetura do sistema; Mecanismos de controle de acesso, Firewalls; Sistemas de detecção de intrusão (IDS); Enumeração de sistemas (nome de usuários e de grupos, tabelas de roteamento, informações de SNMP);
Intranet	Protocolos de rede em uso; Nomes de domínio interno; Blocos de rede; Endereços IP; Serviços TCP e UDP executados em cada sistema identificado; Arquitetura do sistema; Sistemas de detecção de intrusão (IDS); Enumeração de sistemas (nome de usuários e de grupos, tabelas de roteamento, informações de SNMP);
Acesso Remoto	Números de telefone analógicos e digitais; Tipo de sistema remoto; Mecanismos de autenticação;
Extranet	Origem e destino dos pacotes; Tipos de conexão; Mecanismos de controle de acesso;

O passo a passo do *footprinting*:

- Determinar o escopo do *footprinting*. Toda a organização? Apenas a matriz? Ou seja, qual o perímetro de seu alvo.
- Busca de informações de domínio público:
 - o Sites Web da organização;
 - o Buscas na Internet
 - o Buscas na USENET
 - o Endereços
 - o Números de Telefones
 - o Nomes e e-mails de funcionários
 - o Normas de privacidade ou segurança que indiquem o tipo de mecanismo de segurança utilizado
 - o Links para outros servidores relacionados à organização
- Enumeração de redes
 - o Buscas na Faperj
 - o Buscas no Internic
 - o Servidores de Whois

- o Quem registrou o domínio
- Contato admini
- o Quando o registro foi criado
- o Os servidores de domínio e outros domínios relacionados

Mail Transfer

- Determinando a topologia da rede
 - o Usando o traceroute
 - o Identificando firewalls
 - o Usando VisualRoute, cheops, scotty

4.2 Varredura

Se o *footprinting* é o equivalente a cercar o um lugar em busca de informações, a varredura (ou *scanning*) é equivalente a bater nas paredes para descobrir todas as portas e janelas. O passo seguinte então é determinar quais sistemas estão ativos e alcançáveis a partir da Internet, usando uma série de ferramentas e técnicas, como, por exemplo, varreduras de ping, varreduras de porta e ferramentas de descoberta automatizadas. É importante lembrar que um sistema (um endereço IP) descoberto utilizando o *footprinting* anterior não significa que esta máquina está ligada, ou mesmo se ela existe, por isso a varredura é necessária, ela que vai determinar os seus alvos. Vamos citar alguns métodos de varredura:

- Varreduras de ping de rede: fping, gping, hping, nmap -sP, Pinger, Ping Sweep etc.
- Consultas ICMP: icmpquery, icmpush
- Varredura de Portas:
 - o Varredura TCP
 - o Varredura TCP SYN
 - o Varredura TCP FIN
 - o Varredura TCP Árvore d Natal
 - o Varredura TCP nula
 - o Varredura UDP
 - o Para sistemas UNIX/Windows use: nmap, strobe, udp_scan, Netcat
 - o Para sistemas Windows use: PortPro, PortScan, Stealth
- Detecção de Sistema Operacional: nmap -O, queso
- Pacotes completos: scotty, cheops e ferramentas de gerência.

4.3 Enumeração

Existem muitas maneiras de se extrair contas válidas e recursos exportados de sistemas, um processo que chamamos de enumeração.

A principal diferença entre técnicas de coleta de informações vistas e a enumeração está no nível de invasividade – a enumeração envolve conexões ativas e consultas dirigidas que podem (e deveriam!) ser registradas pelo sistema alvo.

O tipo de informação enumerada por atacantes pode ser classificada, de forma geral, nas seguintes categorias:

- Recursos de compartilhamento de rede;
- Usuários e grupos;
- Aplicativos e suas versões (Ex. através de “*banners*”)

No Windows NT:

- Use o net view. Ex:
 - o net view /domain
 - o net view /domain:lab123
- Use o NTRK (Windows NT Resource Kit)
- Use o Legion, NAT
- Ferramentas para SNMP específicas do NTRK (Ex. snmputil)
- Enumeração de *banners*, usando telnet para portas específicas

No Unix:

- showmount -e
- rpcinfo -p
- finger -l @vitima.com.br
- rusers, rwho, etc.
- Telnet para o servidor de email a fim de identificar logins;
- Enumeração de *banners*, usando telnet para portas específicas;
- Use o Netcat
- Use o SamSpade

4.4 Ferramentas utilizadas por *hackers*

A maioria das ferramentas utilizadas por hackers pode se encaixar em um das seguintes grupos:

4.4.1 Dispositivos Destrutivos

- Emails bomba
- Ferramentas de DoS/DDoS
- List Linking (cadastramento de uma pessoa em milhares de listas de discussão)
- Vírus

4.4.2 Trojans

- Código não autorizado dentro de um programa legítimo;
- Programa (trojan) simplesmente renomeado para o nome de um executável conhecido;

- Alguns trojans não são destrutivos, eles de alguma forma coletam informações do sistema/usuário e transferem para o atacante;

4.4.3 Password Crackers

- Ferramentas para quebra de senhas. Ex:
 - o John the Ripper
 - o Cracker
 - o L0phtcrack
 - o NTcrack
 - o Zipcrack
 - o Netcrack
 - o PGPcrack

4.4.4 Scanners

- Ferramentas para varredura de redes:
 - o Nessus
 - o Nmap
 - o NSS
 - o Strobe
 - o SATAN
 - o SAINT
 - o Internet Security Scanner - SafeSuite (ISS)
 - o Cybercop (NAI)
 - o Network ToolBox
 - o Stealth

4.4.5 Sniffers

- Ferramentas de análise de protocolos e de captura de pacotes na rede:
 - o Sniffer (NAI – para Windows)
 - o LinSniff (para Linux)
 - o SunSniff (para Sun)
 - o Snoop (Sun)
 - o Tcpdump
 - o Snort

4.5 **Exploits e BufferOverflows**

Um *exploit* é um programa utilizado para explorar uma vulnerabilidade de outro programa. Não existe um programa chamado *exploit*, existem exploits para explorar vulnerabilidades específicas de softwares específicos, e mais, de versões específicas. Ou seja, você vai encontrar, por exemplo, um exploit para o Servidor de DNS (bind) versão 4.2 , versão para RedHat, Slackware, etc. Às vezes os exploits suportam várias versões, mas isso não é comum.

Por que os exploits são tão específicos? Para responder esta pergunta temos que aprender algo sobre *buffer overflows*, nem sempre os exploits estão baseados em *buffer overflows*, mas estão baseados em, digamos 90% dos casos.

Este assunto é um tanto complexo, e é uma das técnicas de hacking mais avançadas. Envolve dominar bem as linguagens de programação, e até mesmo em baixo nível (assembly). Vamos tentar dar uma explicação com menos detalhes técnicos para o assunto, para não fugir ao escopo deste curso.

Buffer Overflow é um termo bem conhecido no mundo da segurança, portanto você deve se esforçar para entendê-lo. Quando vc está usando um determinado software, seja diretamente na sua máquina, ou indiretamente (interagindo com um servidor) você normalmente insere dados no programa e recebe respostas corretas? Ok, vamos nos concentrar no ato de inserir dados em um programa.

O que aconteceria se você colocasse dados de formas incorretas ou uma quantidade enorme de dados? Bem, se o programa não estiver bem escrito, ele vai travar, ou mesmo apresentar erros absurdos.

Se um programa apresentar tal comportamento provavelmente ele será vulnerável a um *buffer overflow*! Um *buffer overflow* ocorre quando, ao inserir uma quantidade de dados maior do que o programa está preparado para tratar (estourando seu *buffer*) ele irá começar a executar instruções que não estão programadas. Assim, você é capaz de colocar instruções nos próprios dados que está inserindo, fazendo o computador executá-las!

Se o programa em questão tiver privilégios no sistema (imagine remotamente) então você conseguirá inserir instruções com o mesmo privilégio do programa, podendo realizar ações como administrador ou super-usuário.

Descobrir falhas deste tipo e escrever exploit é uma tarefa bastante complexa, e apenas hackers em nível avançado possuem tal expertise. Criar programas que não sejam vulneráveis também requer bom nível de conhecimento de segurança na programação.

Para proteger seus sistemas contra o uso de tais técnicas a melhor coisa a fazer é ficar ligado nos últimos exploits nos sites de hackers e nos *advisories* dos fabricantes.

5. Contramedidas:

5.1 Contramedidas Básicas

O primeiro ponto a se considerar na implementação de contramedidas é considerar todas as opções possíveis. Por exemplo, no que diz respeito à conexão com a Internet: Você poderia simplesmente não usar a conexão? Você poderia optar por outro provedor de serviço (ISP ou *backbone*)? Uma conexão com a Internet representa muitos riscos, então, é necessário estudar bem as opções existentes em cada caso, não só no que se refere ao provedor, mas na forma como sua rede estará conectada, por exemplo, você utilizará muitos endereços Internet, ou NAT?

Depois de todas opções estudadas e decisões tomadas, agora podemos pensar em contramedidas específicas para os diversos pontos críticos da rede:

- Hosts – Melhorar a segurança das máquinas (todas), isto inclui:
 - o Utilização de antivírus e a política de atualização do mesmo;
 - o Varredura de toda a rede, utilizando *scanners* de vulnerabilidades;
 - o Instalar IDS baseados em *host* nas máquinas mais importantes;
 - o Implementar geração e análise de *logs* dos principais sistemas;
 - o Implementar uma boa política de senhas;
 - o Implementar uma política de backup para os servidores;
- Reforçar o perímetro da rede – Identificar corretamente os domínios separados, às vezes chamado “interno” e “externo”, verificar a existência de pontos de entrada na rede: como pode alguém de fora acessar recursos que estão dentro do perímetro? As políticas de segurança de rede deverão incluir procedimentos para o controle do tráfego entre todos os pontos de entrada. Até mesmo a defesa de perímetro mais rígida pode se mostrar inútil diante do ataque de um modem de discagem sem controle que está conectado à rede.
- Utilização de VPNs;
- Utilização de Assinaturas Digitais;
- Utilização de mecanismos de autenticação físicos (biométicos, etc.);

5.2 Segurança no nível do cliente

O capítulo anterior tratou de alguns aspectos contramedidas de segurança que devem ser executados (em geral) pelo administrador do sistema. Neste capítulo vamos tratar de algumas contramedidas e proteções que o próprio usuário pode utilizar.

5.2.1 Segurança do Correio Eletrônico (E-

É muito fácil forjar um e mail;

O serviço SMTP não possui confiabilidade, autenticidade, não repúdio e nem

- Existem problemas de inicialização automática de aplicativos através do MIME; O sendmail (UNIX) é um dos servidores mais utilizados, e é o que apresenta a

Usar certificados digitais PEM (Privacy-

- Usar o PGP (Pretty Good Privacy)
Usar S/MIME (RSA Data Security)

- Não executar, de forma alguma, programas enviados em anexo;

5.2.2 Segurança dos serviços de terminal

- Não usar telnet, rlogin, rsh e afins;
- Usar o SSH;
- Atenção especial deve ser tomada no ato da conexão, caso o programa informe alteração da chave do servidor. Muitas técnicas, usando um programa chamado *dsniff* estão sendo utilizadas para interceptar sessões de ssh.

5.2.3 Segurança no acesso a Web

- Usar conexão segura (SSL) sempre que possível;
- Desabilitar javascripts;
- Nunca executar “cegamente” programas baixados de páginas Web.

O SSL é realmente seguro?

No final de 1995 havia muita publicidade com relação à segurança do Netscape, em particular, do SSL. Algumas publicações especializadas anunciaram a morte para o comércio na Internet, e muitas outras falavam dos eventos como um tremendo golpe à segurança das comunicações na Internet. No meio de todas essas publicidades, alguns fatos foram distorcidos entre os incidentes, e determinados aspectos do problema ganharam proporções catastróficas.

Na verdade, houve três incidentes relativos à segurança do SSL, e o que ficou provado foi que as chaves de 40 bits não eram seguras, pois não são longas o bastante, e posteriormente que o gerador da chave de sessão do Netscape era ineficiente. Todos estes problemas já estão resolvidos há bastante tempo.

É importante observar que nenhum dos três incidentes compromete o SSL – os fundamentos do protocolo permanecem incólumes. Entretanto, o terceiro incidente constitui um reflexo de uma decisão de programação ineficiente por parte do Netscape. Embora o problema tenha sido solucionado rapidamente em versões subsequentes do Netscape, ele comprova como pequenos erros podem levar a problemas imensos.

5.3 Configuração segura de serviços

Vamos abordar aqui algumas *guidelines* para configuração segura de serviços. Vamos tratar separadamente ambientes UNIX e Windows:

5.3.1 Ambientes UNIX

- *Patches*: verificar os últimos *patches* dos serviços que estão disponíveis. Ex: DNS, E-mail, Web, etc. Manter uma tabela atualizada dos serviços em cada host, com as versões, datas de atualização e site do fabricante.
- Segurança de rede: Permitir o acesso somente aos serviços estritamente necessários. O ideal é bloquear o acesso a todos os serviços e dar permissão somente aos necessários. Se isto não for possível deve-se bloquear os seguintes serviços:

o echo	7	TCP/UDP
o systat	11	TCP
o netstat	15	TCP
o bootp	67	UDP
o tftp	69	UDP
o link	87	TCP
o supdup	95	TCP
o sunrpc	111	TCP/UDP
o News	144	TCP
o snmp	161	UDP
o xdmcp	177	UDP
o exec	512	TCP
o login	513	TCP
o shell	514	TCP
o printer	515	TCP
o biff	512	TCP
o who	513	UDP
o syslog	514	UDP
o uucp	540	TCP
o route	520	UDP
o openwin	2000	TCP
o NFS	2049	UDP/TCP
o X11	6000 até 6000 + n	TCP (n é o número de X servers)
- Ao usar um servidor NFS você estará confiando inteiramente na segurança deste servidor para manter a integridade dos sistemas montados. Muito cuidado deve ser tomado ao configurar o arquivo que exporta os sistemas de arquivo (geralmente `/etc/exports`).
- Terminais seguros: Não permita que o super-usuário se conecte diretamente pela rede no sistema. Edite: `/etc/ttys`, `/etc/security` ou `/etc/default/login`.
- `inetd.conf`: Este é o arquivo de configuração do *inetd*. O *inetd* é conhecido como o super server, através dele é possível executar vários serviços. Muitos serviços são configurados para rodar através do `inetd` (comportamento padrão). Ex: telnet, ftp, talk. Estes serviços podem estar habilitados no `inetd.conf` ou não. Para desabilitá-los basta editar o arquivo e colocar uma cerquilha “#” no início da linha que corresponde a serviço, você estará comentando a linha (as linhas que iniciam com a cerquilha significam um comentário em quase todos os arquivos de configuração do UNIX).
- *Sendmail*: Não use. Use o *postfix* ou o *qmail*.

- Servidor Web: Mantenha sempre a última versão do *Apache Web Server*.
- Senhas: Verifique se está sendo utilizado o password shadow. Edite o arquivo */etc/passwd*, não deve ser possível ver os criptogramas referentes às senhas dos usuários. Apenas no */etc/shadow* ou */etc/master.passwd* deve ser possível vê-los. Utilize programas como o Cracker ou o John para tentar quebrar periodicamente as senhas dos usuários.
- NIS ou NIS+: Não use se você realmente não precisa. Use NIS+ se possível.
- Utilize um *Port Scanner* para verificar as portas que ainda existem abertas no sistema. Lembre-se de verificar UDP também.
- Utilize um Scanner de vulnerabilidades para se certificar de que não esqueceu algo. Ou algum detalhe passou despercebido.
- ATENÇÃO: Não confie nas ferramentas de *scanner*, utilize-as apenas para confirmação e no final do processo, e não para montar seu *checklist*.

5.3.2 Ambientes Windows

- *Patches*: verificar os últimos *patches* dos serviços que estão disponíveis. Ex: DNS, E-mail, Web, etc. Manter uma tabela atualizada dos serviços em cada host, com as versões, datas de atualização e site do fabricante.
- Desativar o compartilhamento de arquivos e impressoras no Windows9x;
- Utilizar o *System Policy Editor*, mas conhecido como *Policy* (poledit.exe), e pode ser encontrado na maioria dos CDs do Windows 98 em *\tools\reskit\netadmin\poledit* ou no site da Microsoft.
- Não instalar a rede dial-up ou desabilita-la no Policy.
- Não instalar o serviço de Registro Remoto, se utilizar, escolher uma boa senha;
- Não habilitar SNMP nos Windows NT/2000;
- Extremo cuidado com arquivos executáveis de terceiros, problemas em potencial: BO e NetBus. No caso de ser infectado (ou não) usar o The Cleaner (<http://www.moosoft.com/cleaner.php3>).
- Utilizar senhas na BIOS dos computadores;
- Bloquear as portas 135-139 no Roteador ou Firewall;
- Utilizar *switches* nas redes, ao invés de *Hubs*;
- Aplicar os *services packs* e *hotfixes* mais recentes;

- Visitar periodicamente <http://www.microsoft.com/security>

5.4 Resposta a ataques e problemas de segurança

Mesmo com sua rede super protegida os seguintes eventos ainda podem ocorrer:

- Ataques e invasões (incidentes);
- Um problema ou uma vulnerabilidade detectada;

No primeiro caso pode haver ataques rotineiros (com os quais você não deve perder muito tempo em se preocupar) ou ataques maciços (Ex. DDoS), podem haver incidentes mais graves, como a detecção de uma invasão, e ataques que são persistentes. Nos últimos casos é necessária intervenção do administrador.

Atenção: Lembre-se da nossa definição de ataque (Ataque é uma tentativa, e pode ter sucesso ou não).

O segundo caso é o melhor pois espera-se que você foi o primeiro e único a detectar tal situação, neste caso você deveria consertar o problema e informar ao fabricante/desenvolvedor. Se não for possível consertar o problema deve-se parar o serviço e informar imediatamente ao fabricante e aguardar a solução.

5.4.1 Resposta a ataques e incidentes

No caso dos ataques mais graves e invasões, temos que estar preparados para tais situações e ter um plano de contingência. Assim como o corpo de bombeiros se comporta, nós devemos nos comportar.

Como agir em um incidente? Esta é a grande questão.

O planejamento começa com a preparação das ferramentas que irão auxiliar na auditoria e na correta descrição do problema. Podemos citar:

- Sistema de detecção de intrusão;
- Sistemas de verificação de integridade;
- Sistemas de geração e armazenamento de logs;
- Sistemas de controle de tempo (correto sincronismo do horário entre os hosts)
- Potes de Mel;

Pontos a serem observados na caracterização de um incidente:

- Horário de utilização atípico;
- Padrões de utilização e erros atípicos;
- Presença de contas novas, ou desconhecidas;
- Utilização de uma conta anteriormente inativa;
- Modificações inexplicadas em arquivos;

- Lacunas nos logs do sistema;
- Descoberta de utilitários *hackers*;

Planejamento de Respostas:

- Inicialmente definir a estratégia e o tempo de trabalho necessário:
 - o Isolar o intruso fora da rede;
 - o Permitir que o invasor continue seu ataque, monitorando-o;
- Gerenciando as informações relacionadas ao incidente:
 - o Data e Hora;
 - o Informações sobre os contatos no suporte ao incidente;
 - o Detalhes técnicos;
 - o Conteúdo da comunicação com outras pessoas;
 - o Manter a gerência informada;
 - o Manter as informações sigilosas em caráter confidencial;
 - o Escolher com inteligência os canais de comunicação;

5.4.2 Resposta a problemas de segurança

O que fazer se você encontrou um problema de segurança ou uma vulnerabilidade? Você pode consertar o *bug* e seguir em frente ou pode decidir reportar a vulnerabilidade para o fabricante.

Segue uma lista de fabricantes e/ou desenvolvedores com seus respectivos contatos, que você pode usar para reportar uma vulnerabilidade:

Allaire	mgin@allaire.com
Alt-N	issues@altn.com
Apache	security@apache.org
Debian	security@debian.org
BSDI	problems@bsdi.com
Caldera	security@calderasystems.com
Checkpoint	cpsupport@ts.checkpoint.com
Cisco	security-alert@cisco.com
Cobalt	security@cobalt.com
FreeBSD	security-officer@freebsd.org
Gordano	support@gordano.com
HP	security-alert@hp.com
IBM	security-alert@austin.ibm.com
IpSwitch	dkarp@ipswitch.com
ISC BIND	bind-bugs@isc.org
KDE	submit@bugs.kde.org
Lotus	security@lotus.com
Microsoft	secure@microsoft.com

NetBSD	security-officer@netbsd.org
Novell	frank@novell.com , ncashell@novell.com
OpenBSD	deraadt@openbsd.org
Qualcomm Qpopper	qpopper@qualcomm.com
Qualcomm Eudora	eudora-bugs@qualcomm.com
Red Hat	bugs@redhat.com
SCO	security-alert@sco.com
Slackware	security@slackware.com
SGI	security-alert@sgi.com
Sun	security-alert@sun.com
SuSE	security@suse.de
TurboLinux	K8e@turbolinux.com
WarFTPd	jgaa@jgaa.com
Wu-FTPd	wuftp-members@wu-ftp.org

Para reportar problemas de segurança para o público a melhor forma são as listas de discussão, a BUGTRAQ, uma das mais importantes pode ser encontrada em:
<http://www.securityfocus.com>

Para se cadastrar na BUGTRAQ:

Envie um email para:

listserv@securityfocus.com

Deixe o campo *subject* do email vazio.
Preencha o corpo da mensagem com:

SUBSCRIBE bugtraq André Barbosa

Utilize o seu primeiro e segundo nome. Para saber mais sobre a BUGTRAQ leia a FAQ em: <http://www.securityfocus.com/frames/?content=/forums/bugtraq/faq.html>

Você também deve reportar para o CERT: <http://www.cert.org>

6. Ferramentas de Segurança:

6.1 Controle de acesso

Esta seção engloba quaisquer dispositivos de controle de acesso à rede, sistemas e até mesmo físicos. Podemos citar:

- o Controle de acesso físico: Uso de smart cards, senhas e chaves eletrônicas;
- o Controles de acesso biométricos à sistemas: *fingerprint*, íris, etc.

6.2 Criptografia

A criptografia é uma das armas mais poderosas da segurança. Ela é, na maioria das vezes, utilizada indiretamente através de ferramentas, protocolos e sistemas específicos. Ex.:

- IPsec
- Assinaturas Digitais
- Verificadores de Integridade (Funções de Hash, Ex. MD5)
- PGP
- Criptografia de Senhas (Crypt3 + DES)
- SSH
- SSL

6.3 Firewalls e IDS

Os Firewalls e os Sistemas de Detecção de Intrusão são dois componentes principais que não devem faltar na infraestrutura de segurança de uma rede, temos então, capítulos específicos para eles nesta apostila.

6.4 Scanners

São ferramentas que detectam automaticamente fraquezas na segurança em *hosts* remotos ou locais.

Como eles funcionam?

Os *scanners* (também conhecidos como *port scanners*) interrogam várias portas TCP/IP do alvo e armazenam os resultados., eles conseguem assim várias informações, tais como:

- Quais serviços estão rodando atualmente;
- Quais são os usuários que rodam estes serviços;
- Se login como usuário anônimo é suportado;
- Se alguns serviços de rede requerem ou não autenticação;

Muitos dos scanners citados a seguir não são apenas scanners de porta, possuem também como opções adicionais funções tais como:

- Verificação do sistema operacional;
- Verificação de diversas vulnerabilidades conhecidas;

Exemplos:

- o Nessus
- o Nmap
- o NSS
- o Strobe
- o SATAN
- o SAINT

- o Internet Security Scanner - SafeSuite (ISS)
- o Cybercop (NAI)
- o Network ToolBox
- o Stealth

6.5 Sniffers

Sniffers são dispositivos que capturam pacotes na rede. Seu propósito legítimo é analisar o tráfego da rede e identificar áreas potenciais de problemas ou interesse.

Os *sniffers* variam muito em funcionalidade e design, alguns analisam apenas um protocolo, outros podem analisar centenas. Em geral, *sniffers* modernos analisam ao menos os seguintes protocolos:

- Ethernet padrão;
- TCP/IP;
- IPX;
- DECNet;

Nota sobre programas de captura de teclas:

Sniffers diferem de programas de captura de teclas pois capturam pacotes em uma rede, enquanto os outros (Key-capture) capturam apenas o que foi digitado num teclado de um único terminal.

Exemplos:

- o Sniffer (NAI – para Windows)
- o LinSniff (para Linux)
- o SunSniff (para Sun)
- o Snoop (Sun)
- o Tcpdump
- o Snort

6.6 Password Crackers

Os *password crackers* são programas que quebram a segurança das senhas revelando o conteúdo previamente criptografado. Para um *password cracker* funcionar corretamente é necessário:

- O arquivo de senhas (com os criptogramas), ou alguns criptogramas das senhas;
- Uma máquina com um processador veloz;
- Um dicionário de senhas;
- Tempo;

A maioria destes programas funciona através do método conhecido como força bruta. A força bruta funciona da seguinte forma:

Cada palavra de um dicionário é criptografada e o resultado (criptograma) é comparado ao criptograma da senha que se deseja quebrar, quando ocorrer uma coincidência, pronto, a senha será revelada.

Notamos então, que não é correto dizer que um programa destes decifra uma senha (muitas pessoas dizem), eles não decifram o criptograma e não fazem criptoanálise (salvo algumas exceções) pois são baseados na força bruta, ou seja, na comparação.

Exemplos:

- o John the Ripper
- o Cracker
- o L0phtcrack
- o NTcrack
- o Zipcrack
- o Netcrack
- o PGPcrack

7. Firewalls:

7.1 Tipos de Firewall

Um firewall é qualquer dispositivo destinado a prevenir atacantes externos de acessar sua rede. Este dispositivo pode ser um computador, um roteador, ou um hardware dedicado. Firewalls também podem analisar pacotes de vários protocolos e processar os pacotes através de regras que irão permitir ou não a passagem deste pacote.

Podemos classificar os firewalls em dois grandes grupos:

➤ **Nível de Rede (Filtro de Pacotes)**

Firewalls no nível de rede são tipicamente roteadores com poderosas funções de filtragem de pacotes que podem permitir ou bloquear os pacotes em ambas as direções, baseados no:

- o Endereço de origem
- o Protocolo
- o Números de porta
- o Conteúdo de pacote
- o Flags TCP
- o E outras características dos protocolos envolvidos

➤ **Gateways de aplicação (proxies)**

Os proxies são uma forma completamente diferente de controle de tráfego. Nos filtros de pacotes, os usuários não precisam se preocupar com a existência do Firewall, ele é transparente, com os proxies isso não acontece, os proxies agem de tal forma que ficam no meio da conexão, aceitando requisições dos clientes

e abrindo conexões com os destinos, sendo intermediários neste processo. Isso faz com que os proxies sejam específicos para cada aplicação. Temos proxies de HTTP, FTP, SMB. etc. Existem também os proxies genéricos e os proxies de circuito Ex. SOCKS.

Firewalls modernos podem realizar ambas as funções descritas anteriormente e ainda prover outras (através de módulos, ou não), tais como:

- Criptografia
- NAT (Network Address Translation)
- Autenticação

7.2 Arquiteturas

Ocasionalmente, as empresas preferem implementar um firewall baseado apenas em uma máquina, seja ela um host ou um roteador. Com frequência, porém, os firewalls mais rigorosos são compostos de várias partes. Vamos examinar alguns exemplos de arquiteturas empregados:

- Roteador com listas de acessos
- Gateways simples
- *Screened host firewall (single-homed bastion)*
- *Screened host firewall (dual-homed bastion)*
- *Screened subnet firewall system*

7.3 Fabricantes

Alguns Firewalls e seus fabricantes/desenvolvedores:

- Checkpoint FW-1 (<http://www.checkpoint.com>)
- Cisco PIX (<http://www.cisco.com>)
- Aker (<http://www.aker.com.br>)
- Ipfiler (Darren Reed)
- PF (OpenBSD team) <http://www.openbsd.org>

7.4 Testando a segurança de um firewall (*penetration test*)

Um teste de invasão tem por objetivo verificar a resistência do sistema em relação aos métodos atuais de ataque. Este método pode ser simplesmente um tipo de engenharia social, onde alguém do *Tiger Team* liga para alguns dos funcionários e pergunta pela identificação do usuário e senha ou mais complexo utilizando técnicas de *buffer overflow* para ganhar acesso de root.

Diariamente são descobertos novos furos nos mais variados sistemas, por isso é de fundamental importância que o *Tiger Team* utilize técnicas reais, pois caso isso não ocorra o teste pode tornar-se inválido.

Após o teste temos duas possibilidades:

- 1.O Tiger Team obtém êxito, logo o sistema de segurança está falho.
- 2.O Tiger Team não obtém êxito, logo o sistema de segurança está adequado.

A segunda afirmativa pode não ser verdadeira uma vez que seu sistema foi submetido a uma equipe que está sujeita a erros.

7.4.1 Testando o Firewall

O Firewall é um sistema, e qualquer sistema pode ser ultrapassado. Na verdade várias instituições instalam Firewall com o objetivo de se protegerem da Internet, depois eles mesmo criam regras (brechas) para permitirem conexões com vendedores e parceiros. Além disso, muitos Firewalls são instalados com a configuração básica e nunca são testados para verificar sua eficiência. Um bom teste de invasão consegue identificar os buracos de forma que você saberá que eles existem.

7.4.2 Identificando as vulnerabilidades de alto risco:

Um teste de invasão pode dar uma boa idéia sobre as vulnerabilidades de alto risco presente em seu sistema. Financeiramente não é interessante utilizar o teste de invasão para identificar todas as possíveis vulnerabilidades do seu sistema. Se seu objetivo é simplesmente identificar as vulnerabilidades, considera a utilização de uma ferramenta de SCAN, como: nmap, SATAN, ISS scanner. Lembre-se, você está pagando por um teste de invasão pelo conhecimento e expertise do *Tiger Team* além da habilidade em explorar as vulnerabilidades. As ameaças e vulnerabilidades que não conseguirem explorar podem ser identificadas pela ferramenta de *scan*, pense nisso, antes de contratar um teste de invasão.

7.4.3 Riscos envolvidos em um teste de invasão :

Um importante aspecto referente ao teste de invasão é que ele pode gerar uma falsa sensação de segurança. A idéia de que "Eles fizeram o melhor que podiam e não obtiveram êxito" não é válida.

Sua rede pode ter vulnerabilidades que o *Tiger Team* não tenha encontrado ou talvez elas não existam no momento do teste, mas podem vir a existir após alguma mudança na configuração da rede.

Um importante ponto a se lembrar é, "Você terá o que você pagou". Existem algumas pessoas que executam teste de invasão e acham que seu trabalho é entrar na rede - sem identificar as vulnerabilidades. É recomendável que a pessoa que vai realizar o teste de invasão apresente um documento detalhando exatamente quais são os resultados finais a serem alcançados. Além disso você deve considerar a realização do teste de invasão em vários momentos do seu sistema.

7.4.4 Danos ao sistema durante o teste :

Fatores que devem ser levados em consideração são os possíveis danos causados ao sistema uma vez que este pode ser afetado ou arquivos importantes podem ser perdidos - é importante a definição de parâmetros que identificam os pontos onde o teste tem validade. Você pode ignorar qualquer tipo de DoS (Denial Of Service) conhecido. As pessoas que realizam o teste de invasão, por definição, não são usuários legítimos. O bom teste de invasão não pode ficar preso a um único aspecto do seu sistema. Muitas pessoas que realizam teste de invasão requerem que seja indicado um departamento que assuma as responsabilidades no caso de ocorrer algum dano ao sistema.

Enquanto você não tiver testado sua ferramenta de IDS ou montado seu plano de resposta a incidentes, a pessoa responsável pelo teste não deve receber nenhum tipo de informação sobre o seu sistema, parceiros comerciais. Isso pode protegê-lo quanto a validade do teste de forma que intrusos verdadeiros não tenha acesso a estas informações.

Os testes devem ser conduzidos utilizando-se ferramentas previamente definidas. Enquanto os pontos acima não forem atendidos, o responsável pelo teste não deve ter acesso a sua rede.

Informações públicas, tais como : estrutura da empresa, lista de telefones internos, entre outras podem ser passadas para pessoa que vai realizar o teste, justamente para ganhar tempo.

A pessoa que vai realizar o teste não deve violar a privacidade e os direitos individuais.

Lembre-se que trata-se de um teste para avaliar o sistema e não as informações privadas. Todos os dados coletados, incluindo logs, arquivos, senhas e qualquer outro tipo de informação obtida deve ser devolvidas a instituição sem que cópias sejam retidas pela organização que realizou os testes.

Todos os testes devem ser realizados de forma instrutiva.

Qualquer tipo de teste que possa causar danos ao sistema deve ser realizado em períodos de baixa ou sem atividades.

Um relatório detalhado deve ser entregue contendo todos os passos executados mostrando onde ganhou acesso e onde não. O relatório deve conter recomendações detalhadas para a correção de qualquer vulnerabilidade encontrada.

7.4.5 Fase-1: Coleta de dados e planejamento:

Nesta fase o *Tiger Team* vai aprender tudo que puder sobre o alvo e não estará necessariamente preocupado com vulnerabilidades do sistema. Ele tentará obter informações sobre a estrutura da diretoria, número dos telefones/ramais, relação dos parceiros, dos vendedores, ou seja, todo tipo de informação.

Raramente a pessoa que realiza o teste tem que recorrer a mecanismos como *buffer overflow*. A relação de telefones possui muitas informações úteis para o invasor.

Outro tipo de informação importante é a topologia da rede. Conforme a topologia o invasor pode determinar quais os pontos mais vulneráveis.

Durante esta fase um detalhado plano de ataque é construindo, eu pessoalmente chamo esta fase de aproximação indireta, um termo utilizado em estratégias militares e planejamento de atos de terrorismo.

7.4.6 Fase 2: Pesquisa do sistema

Nesta fase ainda dentro do que chamo aproximação indireta podemos colher mais informações sobre a instituição que está sendo testada. Isso inclui a consulta ao whois via web: FAPESP (www.fapesp.br), ao InterNIC (www.inetnic.net) e ao ARIN (www.arin.net), além destes existem outros servidores de whois.

Com estas informações temos uma idéia de que a instituição está na rede e de seus IPs.

7.4.7 Fase 3: Testes do sistema

Agora iniciamos a fase que chamo de aproximação direta, onde temos os seguintes passos:

A) Identificar o caminho para acessar a instituição, podemos fazer isso com o traceroute, mas em vez de usar o traceroute tradicional, podemos usar uma ferramenta especial para fazer um traceroute a uma porta TCP ou UDP específica, desta forma conseguimos burlar os filtros de ICMP. Este fase permite compreender o caminho de acesso a instituição, ou seja, nos dá uma visão lógica do caminho.

Nosso objetivo é determinar o caminho e as ACLs implementadas nos roteadores e Firewalls. Para tanto podemos usar a ferramenta firewalk. Esta é uma ferramenta muito interessante que permite determinar as ACLs implementadas, ou seja, identificar quais serviços são permitidos através da ACL.

B) Agora seria uma boa idéia é verificar que tipo de informação podemos recuperar do servidor de DNS. Se o servidor estiver mal configurado é possível fazer uma transferência de zona, isso nos dá muitas informações úteis, um exemplo é a recuperação do registro HINFO, se esta informação estiver disponível, saberemos exatamente o tipo de sistema operacional da instituição. Podemos usar vários comandos diferentes para este fim como nslookup, dig e host. Um dos objetivos é determinar o endereço do firewall para que possamos testá-lo. Podemos fazer uma análise destas informações e rapidamente com o auxílio do grep podemos descobrir todas as máquinas que possuem a palavra teste em seu nome, se estas máquinas estiverem mal configuradas é um bom local para tentar um acesso não autorizado, da mesma forma podemos usar o grep para identificar outros padrões de nome como linux, sun,bsd, etc.

C) Após identificar a topologia da rede, precisamos determinar quais os hosts que estão ativos e conectados a Internet. Podemos usar para tal fim ferramentas como nmap e o fping. O nmap permite analisarmos uma determinada faixa de endereços. Enquanto a maioria das ferramentas de ping trabalham em cima de ICMP com o nmap podemos fazer um ping através do TCP, ou seja, se os pacotes ICMP estiverem sendo filtrados no roteador de borda, com o nmap podemos achar os hosts ativos usando por exemplo a porta 80.

D) Se uma máquina esta ativa e conectada a Internet, chegou o momento de fazer um port scan. O port scan tem por objetivo determinar quais portas (TCP/UDP) estão ativas. A identificação de portas é fundamental para identificar o sistema operacional e as aplicações em uso. Através dos serviços ativos podemos ganhar acesso a máquinas que estão mal configuradas ou rodam versões de programas com vulnerabilidades conhecidas. Existem várias ferramentas que permitem a realização de port scan: nmap, strobe, tcp_scan, udp_scan, netcat e queso são algumas delas.

E) Após descobertas as portas ativas de cada host conectado a Internet, chegou a hora de obtermos mais informações dos hosts. Isso inclui banner ou qualquer outro tipo de informação.

Informações fornecidas pelos serviços de SNMP, finger, rusers SMTP ou NetBIOS permitem que montemos uma configuração detalhada além de conseguirmos informações sobre os usuários de cada sistema. Agora podemos conectar a cada uma das portas TCP/UDP e analisar as respostas, afim de identificar informações sobre versão e descobrir servidores vulneráveis, mas não é somente a versão que nos interessa, mas informações do sistema, se serviços como finger e ruser estiverem ativos, nós podemos obter informações dos usuários do sistema. Através do SNMP utilizando uma conexão UDP a porta 161 podemos usar *query* com *snmpget*, *snmpwalk* e *snmpstat* para obter algumas informações.

F) Agora que já temos um conjunto de informações sobre os hosts, como máquinas ativas, os serviços que rodam, informações de usuários entre outras, podemos montar um mapa de vulnerabilidades. O objetivo deste mapa é associar as informações do sistema com as vulnerabilidades conhecidas.

Deve-se fazer log detalhado de todos os testes e *scans* bem como de seus resultados. O objetivo do log é permitir que após os testes possa-se fazer uma relação para determinar se os testes causaram algum dano ao sistema e garantir que outro intruso não tenha ganhado acesso ao sistema durante o teste.

12. Conclusão

O teste de invasão deve fazer parte do programa de segurança da sua empresa. Mas deve-se observar os pontos acima tratados, para que se possa tirar um real aproveitamento do dinheiro empregado em tal atividade. Você deve restringir as informações sobre os testes somente aos departamentos competentes, pois alguns funcionários desavisados ou inocentes podem deixar esta informação vazar e alguém pode aproveitar para realizar seu próprios testes.

8. Sistemas de Detecção de Intrusão (IDS):

Baseados nas informações de várias entidades de pesquisa, tais como: CERT e ICSA, podemos afirmar que um sistema foi atacado ou invadido mais de uma vez por segundo no ano de 1999. Só nos EUA a ICSA identificou que, em média, um site ou computador foi invadido a cada 11 minutos.

Estas estatísticas nos levam a enorme necessidade de poder rastrear e identificar estes ataques. O sistema que tem a capacidade de fazer isso é um Sistema de Detecção de Intrusão ou IDS.

Um Sistema de Detecção de Intrusão (IDS - Intrusion Detection Systems) permite a notificação quando da ocorrência de tentativas de intrusão segundo a verificação de certos padrões de ataque que podem ser configurados dependendo da ferramenta que se está utilizando.

Se sua casa possui um sistema de alarmes contra ladrões, você possui um sistema de IDS relativamente sofisticado. Ele pode detectar tentativas de invasão e tomar alguma ação baseado na detecção.

8.1 Tipos de IDS

Os Sistemas de Detecção de Intrusão são normalmente divididos em três grandes grupos. Sendo que, atualmente, existem IDS que possuem características mistas. Vamos listar estes três grupos, ressaltando algumas características importantes:

- Baseados em rede (NIDS)
 - o Sensores
 - o Estação de gerência
 - o Análise de assinaturas
- Baseados em host
 - o Visão localizada
 - o Gera informação de melhor qualidade
- Verificadores de integridade de arquivos
 - o Funções de hash

Vamos ressaltar em especial aqui o IDS de rede, que é um dos mais utilizados. Este tipo de IDS coloca a interface de rede no chamado modo promíscuo, o que permite “escutar” todos os pacotes que trafegam na sub-rede no qual está inserido, semelhante a um *sniffer*. A diferença é que, para cada pacote, o IDS compara seu conteúdo e cabeçalho a um banco de assinaturas de ataques, bem parecido com um anti-vírus. Deve-se lembrar também a necessidade de espelhamento (ou span) de porta se a rede for comutada (switches).

8.2 Fabricantes

Intruder Alert	Axent	www.axent.com	Host
NetProwler	Axent	www.axent.com	Rede
Real Secure	ISS	www.iss.net	Host/Rede
NetRanger	Cisco	www.cisco.com	Rede
NFR	NFR	www.nfr.net	Rede
Session Wall	C.Associates	www.ca.com	Rede
Snort	Marty Roesh	www.snort.org	Rede
Abacus	Psionic	www.psionic.com	Host

Uma tabela comparativa:

Product Comparison Chart						
Criteria	Weight	NFR	RS	SW	NP	NR
Functional Testing						
Attack Set Detection	5	4.0	4.0	2.0	1.0	4.0
Protection of IDS Against IP Desync	3	4.0	1.0	0.0	0.0	N/A
Repetitive Attack Suppression	3	1.0	0.0	4.0	0.0	3.0
Filter Customization	5	4.0	1.0	2.0	3.0	2.0
Alerting	4	3.0	3.0	4.0	3.0	3.0
Logging	4	4.0	3.0	3.0	3.0	4.0
Reporting	3	2.0	3.0	4.0	3.0	3.0
Distributed Architecture	2	2.0	3.0	1.0	0.0	4.0
Performance Testing						
Engine Speed	3	3.0	3.0	1.0	4.0	N/A
Packet Reassembly	5	2.0	4.0	N/A	0.0	N/A
Efficiency of Filtering	5	2.0	3.0	N/A	2.0	N/A
Quantitative Assessment	Weighted GPA	2.90	2.64	2.41	1.79	2.90
Product Usability						
Interface Usability	3	2.0	3.0	4.0	2.0	2.0
Appliance/OS Implication on IDS	4	4.0	1.0	1.0	1.0	3.0
Maturity of Product	2	2.0	4.0	4.0	1.0	3.0
Product Concept	1	3.0	3.0	4.0	4.0	4.0
Company Focus	3	4.0	4.0	2.0	3.0	1.0
Price	4	4.0	3.0	2.0	3.0	1.0
Overall System	Weighted GPA	3.03	2.69	2.43	1.90	2.84
Key: NFR= Network Flight Recorder 3.0 RS= Real Secure 3.0 SW= SessionWall 3.31 NP= NetProwler 3.0 NR= NetRanger 2.2						
Grading Scale: 4.0 represents the highest score Weight Scale: 5 is the most relevant criteria						

Figure 2 - Product Comparison Chart

8.3 Testando o sistema de detecção de intrusão (IDS)

Infelizmente, devido a grande variedade de vulnerabilidades, detectar uma intrusão em sua rede não é algo simples. É praticamente impossível que uma pessoa detecte uma invasão de rede em tempo real (on-the-fly) e tome alguma ação de imediato.

As ações da sua ferramenta de IDS devem estar diretamente relacionadas com o plano de resposta a incidentes. Na verdade a definição de um plano de resposta a incidentes é um fator tão crítico que caso você não tenha um, provavelmente um teste de invasão não irá ajudá-lo muito. Seu plano de resposta a incidentes deve abranger basicamente os seguintes pontos:

Qual o objetivo do plano de resposta para cada tipo de incidente?

Quais as ações legais existentes?

Serão tomadas ações legais no caso de um incidente?

Que tipo de publicidade (a respeito do ataque) é permitido?

Quem é responsável por conduzir a resposta ao incidente?

Quem fará parte do grupo de resposta a incidente?

Que nível de autoridade é requerido para o grupo de resposta a incidente?

Como você conduz uma resposta a um incidente está diretamente relacionado ao tipo de negócio da sua instituição. Os bancos por exemplo, devem tomar algumas ações junto a federação nacional de bancos.

Após você ter seu plano de resposta ao incidente montado, você pode testá-lo efetivamente e refiná-lo utilizando o teste de invasão. É uma boa idéia anunciar a realização do primeiro teste, uma vez que seu propósito é ajustar seu plano de resposta e verificar se ele funciona.

A alta taxa de falso-positivos ocorre quando a ferramenta classifica uma ação como uma possível intrusão, quando na verdade trata-se de uma ação legítima.

Um falso-negativo ocorre quando uma intrusão real acontece mas a ferramenta permite que ela passe como se fosse uma ação legítima.

Erro de *subversion* ocorre quando o intruso modifica a operação da ferramenta de IDS para forçar a ocorrência de falso-negativo.

Um bom exemplo de falso-positivo é o ataque de SYN FLOOD. O simples fato de acessar um determinado tipo de página pode gerar uma detecção da ocorrência de um ataque SYN FLOOD.

Você certamente não quer que suas páginas fiquem fora do ar a todo momento que um usuário acessar seu site. É muito difícil definir regras que diferenciem entre atividades hostis e autorizadas. O teste de invasão pode ser utilizado com a finalidade de demonstrar efetivamente se sua ferramenta de IDS está operando conforme o esperado e ajudá-lo no refinamento das regras de forma a reduzir a taxa de false-positive.

9. Tecnologias avançadas de segurança:

9.1 Secure Sockets Layer (SSL)

O SSL foi criado pela Netscape e foi adotado pela comunidade e é hoje o padrão para comunicação segura pela Web. O SSL está montado sobre a camada de transporte (TCP) e possui dois conceitos básicos:

- Conexão SSL;
A conexão para o SSL é uma relação ponto a ponto. As conexões são transientes e estão sempre relacionadas a uma sessão.
- Sessão SSL;
É uma associação entre o cliente e o servidor, elas são criadas através de um protocolo de *handshake* específico do SSL (*Handshake Protocol*). As sessões definem um conjunto de parâmetros para os algoritmos de criptografia.

Fases da conexão:

- Estabelecimento de capacidades de segurança;
- Autenticação do servidor e troca de chaves;
- Autenticação do cliente e troca de chaves;
- Término;

Algoritmos utilizados:

- RSA;
- DSS;
- MD5;
- SHA;

9.2 IP seguro (IPsec) e Redes Virtuais Privadas (VPNs)

Como sabemos, o TCP/IP não possui nenhum nível de confidencialidade ou autenticação, a não ser que seja implementado na camada de aplicação.

A proposta do IPsec é prover tais serviços de forma transparente ao usuário, de forma que qualquer aplicação possa utilizar, sem a necessidade de alterar programas clientes ou servidores.

RFCs: 1825, 1826, 1828, 1827, 1829.

O suporte ao IPsec é opcional ao Ipv4 e obrigatório ao Ipv6.

A documentação do IPsec é dividida de forma a abranger os vários componentes do modelo, são estes:

- Arquitetura
- ESP – Encapsulating Security Payload
- AH – Authentication Header
- Algoritmos de criptografia
- Algoritmos de autenticação
- Gerenciamento de chaves
- DOI - Domínio de interpretação

O conceito básico de uma conexão IPsec é a *Security Association* (SA). Deve existir uma SA para cada conexão, e em cada sentido, então, para uma VPN ponto a ponto são necessárias duas SAs.

Os serviços mais importantes do IPsec estão contidos no AH e no ESP, sendo que, no estabelecimento de uma SA deve ser escolhido qual dos dois utilizar. A tabela a seguir nos dá uma descrição dos serviços providos pelo AH e pelo ESP.

O AH oferece basicamente serviços de autenticação, enquanto o ESP oferece serviços de autenticação e criptografia (podem ser usados simultaneamente).

Serviço	AH	ESP(encryption only)	ESP(enc + utent.)
Controle de Acesso	X	X	X
Integridade	X		X
Autenticação	X		X
Rejeição de replay	X	X	X
Confidencialidade		X	X
Fluxo limitado de tráfego		X	X

É importante ressaltar a complexidade do IPsec, isto implica em dificuldades de configuração e de compatibilidade entre sistemas de fabricantes diferentes. Portanto, nunca confie totalmente num fabricante, prefira homologar os produtos antes de comprá-los.

9.3 Certificados Digitais

O desenvolvimento mais importante no trabalho com os algoritmos de chave pública é a assinatura digital. A assinatura digital provê um conjunto de serviços de segurança que seriam impossíveis de se implementar de outra forma.

Os algoritmos criptográficos de chave pública diferem dos convencionais (ou simétricos) pelo fato de possuírem um par de chaves, a chave pública e a privada. Cada participante deve ter uma chave pública (que todos devem conhecer) e uma chave privada (que só ele deve conhecer). Maiores detalhes podem ser encontrados em qualquer curso ou livro básico de criptografia, visto que estes algoritmos são amplamente difundidos.

Com a assinatura digital deve ser possível:

- Verificar o autor, a data e a hora da assinatura;

- Autenticar o conteúdo do texto assinado;
- Verificar a assinatura com terceiros;

Com base nestas propriedades nós podemos formular os seguintes requerimentos para uma assinatura digital:

- A assinatura deve ser função da mensagem sendo assinada;
- Ela deve usar uma informação única do remetente (chave privada);
- Deve ser relativamente fácil produzi-la;
- Deve ser relativamente fácil reconhecer e verificar a assinatura;
- Deve ser computacionalmente impraticável forjar a assinatura;
- Deve ser prático guardar uma cópia da assinatura;

Uma função de hash segura incluída neste esquema satisfaz estes requerimentos.

Estudo de caso: DSS – Digital Signature Standard:

O DSS usa um algoritmo designado para prover apenas a função de assinatura digital, o DAS – Digital Signature Algorithm, ele é baseado no fato de não se conseguir computar facilmente logaritmos discretos. O DSS não utiliza o RSA.

10. Gerência de Segurança

10.1 Plano de Segurança

Criar um plano de segurança é uma tarefa dinâmica que envolve diversas etapas. Uma vez que você tenha definido suas metas, elaborando um plano de negócios para uso na Internet, você poderá criar o programa de segurança a ser usado para cuidar de eventuais riscos. Você deverá se preocupar com os serviços de curto prazo, destinados a usuários finais, de que a sua empresa precisa, além de levar em consideração o uso estratégico, a longo prazo, da Internet.

Estamos falando até agora da Internet, que pode ser a primeira vista o ponto mais crítico, mas temos que nos preocupar com os fatores internos que podem ser bem mais graves. Resumindo:

- Distribuição de autoridades e privilégios;
- Política de senhas;
- Política de backup;
- Uso do Antivírus;
- Educação em segurança (palestras periódicas);
- Análise de Risco
 - o Destruição de Dados

- o Perda de integridade de dados
- o Perda de integridade do sistema ou da Rede
- o Perda da capacidade de acesso ao Sistema ou a Rede
- o Perda de reputação

Você deverá conduzir uma avaliação de risco dirigida, a fim de:

- o Identificar áreas de risco elevado;
 - o Tratar de riscos de segurança não considerados no conjunto de controles que você utiliza;
 - o Obter um consenso quando os diferentes conjuntos de comandos básicos forem contratórios;
 - o Oferecer uma justificativa para utilizar um ou mais controles de segurança quando o interesse em fazê-lo não for muito grande;
- Análise de ameaças
- Implementar os controles de segurança
- Gerenciar e administrar
- o Monitoração e Auditoria;
 - o Resposta a Incidentes;
 - o Resposta a problemas de segurança;
 - o Administração da Segurança;
 - o Alocação de recursos

10.2 Análise da ameaça

A análise da ameaça é um processo de extrema importância no planejamento da segurança. Ela envolve entender perfeitamente o seu sistema e como ele pode ser explorado, mas do que todas as outras etapas, você deve pensar como um hacker pensaria e ainda avaliar os riscos que podem acontecer. Quanto você gastará em contramedidas para cada área?

Entenda as ameaças reais ao sistema e avalie o risco dessas ameaças. É mais fácil entender isso se você puder se basear na experiência do mundo real, com ataques reais contra sistemas semelhantes.

Descreva a diretriz de segurança exigida para defender-se contra as ameaças. Isto será uma série de instruções como: “somente bancos autorizados têm permissão para modificar o saldo nos cartões Plastcash”, ou “todas as transações com Plastcash precisam ter logs de auditoria”.

Projete as contramedidas que imponham a diretriz descrita anteriormente. Estas contramedidas serão uma mistura de mecanismos de proteção, detecção e reação.

10.3 Normas de Segurança

O CERT (<http://www.cert.org>) possui arquivos com várias normas e *checklists* de segurança. Podemos citar os excelentes *checklists* para UNIX e Windows.

A BS7799 é um conjunto de normas de segurança adotado mundialmente. Ela não é gratuita e pode ser adquirida pela internet em: www.c-cure.org.

~ The End ~

Apêndice A – Glossário de Segurança

Access Control:

Controle de acesso. Mecanismo para prover autenticação e privilégio para usuários em ambientes ou sistemas.

Appz:

Termo do mundo do software pirata que se refere a qualquer aplicativo.

Ataque:

Evento que pode comprometer a segurança de um sistema ou uma rede. Um ataque pode ter ou não sucesso. Um ataque com sucesso caracteriza uma invasão. Um ataque também pode ser caracterizado por uma ação que tenha um efeito negativo, Ex: DoS.

Auditoria:

Processo de análise de causas e efeitos de incidentes, análise de *logs*, etc.

Autenticação:

É o processo de se confirmar a identidade de um usuário ou um *host*, esta pode ser feita na camada de aplicação (através de uma senha), ou mais complexa, no esquema desafio-resposta (utilizando algoritmos específicos).

Back door:

É um programa escondido, deixado por um intruso, o qual permite futuro acesso à máquina alvo. Este termo é um sinônimo para um termo mais antigo: *trap door*.

Bug

Uma falha, ou fraqueza num programa de computador. Veja *vulnerabilidade*.

Cavalo de Tróia

Uma aplicação ou código que, sem o conhecimento do usuário realiza alguma tarefa que compromete a segurança de um sistema, em geral, esta aplicação se apresenta ao usuário de forma rotineira e legítima, Ex.: um simples jogo que contém código malicioso que envia os dados do usuário para um e-mail específico.

CERT

The Computer Emergency Response Team – Uma organização dedicada a segurança, seu propósito é dar assistência à redes que foram invadidas ou estão sob ataque. Eles podem ser encontrados em <http://www.cert.org>

Certificação

Existem duas definições para este termo. Primeira, certificação pode se referir para o resultado de uma avaliação com sucesso

Cavalo de Tróia:

Veja Trojan Horse.

CERT:

Organização de segurança – Computer Emergency Response Team. Podem ser encontrados em: <http://www.cert.org>.

Checksum:

Hash criptográfico. Cadeia de caracteres, geralmente de tamanho fixo que pode ser utilizada para representar de forma única, um arquivo. Ex. Suponha um arquivo com 1000 linhas, ele pode ser representado por um checksum de, digamos, 15 caracteres. Se o arquivo for alterado o checksum (hash) irá alterar.

Crack

Programa utilizado para quebrar licenças de outros programas. Também pode se referir a programas utilizados para quebrar senhas.

Cracker

Indivíduo com conhecimentos elevados de computação e segurança, que os utiliza para fins criminosos, destruição de dados ou interrupção de sistemas. Também pode se referir a programas utilizados para quebrar senhas (Ex. *Password Cracker*).

DES:

Algoritmo de criptografia – Data Encryption Standard, hoje obsoleto pelo AES – Advanced Encryption Standard.

Engenharia Social

Técnica utilizada por hackers e crackers para obter informações interagindo diretamente com as pessoas.

Exploit

Programas utilizados por hackers e crackers para explorar vulnerabilidades em determinados sistemas, conseguindo assim, acessos com maior privilégio.

Firewall

Equipamento e/ou software utilizado para controlar as conexões (que entram ou saem) de uma rede. Eles podem simplesmente filtrar os pacotes baseados em regras simples, como também fornecer outras funções tais como: NAT, proxy, etc.

Flood

Sobrecarga (em geral, de pacotes) causada por eventos não esperados que causam lentidão da rede.

Hacker

Indivíduo com conhecimentos elevados de computação e segurança, que os utiliza para fins de diversão, interesse, emoção. Em geral, hackers não destroem dados, possuem um código de ética e não buscam ganhos financeiros. O termo hacker é atualmente adotado pela mídia de forma indiscriminada, se referindo a crackers por exemplo.

Hacking

É o ato de hackear sistemas, não no sentido único de invadir, mas principalmente de descobrir como funcionam, e se possuem falhas.

Hijacking

É o assalto de uma sessão, geralmente TCP/IP. O assalto de sessão é uma forma de obter o controle de uma conexão iniciada por um usuário legítimo. Ao interceptar esta

conexão o “hacker” pode impedir o usuário legítimo de usar o sistema e tomar o seu lugar.

Hole

Um bug ou uma vulnerabilidade.

Intrusion Detection System -IDS

É um Sistema de Detecção de Intrusão, um software responsável por monitorar uma rede ou sistema e alertar sobre possíveis invasões.

Invasão

Caracteriza um ataque bem sucedido.

Lammer

É uma palavra que os hackers utilizam para identificar os indivíduos que se acham hackers, mas estão ainda no estágio inicial de aprendizado.

Phreaking

São os hackers de telefonia, convencional ou celular.

Scanner

Ferramenta utilizada por hackers ou especialistas em segurança que serve para “varrer” uma máquina ou uma rede, em busca de portas abertas, informações ou serviços vulneráveis.

Script Kiddie

É o indivíduo que saiu do estágio de *lammer* mas que só sabe usar as “receitas de bolo”, programas prontos e ainda não entende muito bem o que está fazendo.

Sniffer

Ferramenta utilizada por hackers e especialistas em segurança e de rede que serve para monitorar e gravar pacotes que trafegam pela rede, dependendo do sniffer, é possível analisar vários dados dos pacotes, analisar protocolos, ver dados específicos da camada de aplicação, senhas, etc.

Spoofing

É uma forma de manter uma conexão com uma máquina se fazendo passar por outra na qual ela confie. Um termo muito utilizado é o *IP Spoofing*, que significa o uso de vulnerabilidades do Protocolo TCP/IP que permitem a ação descrita acima.

Trojan, Trojan Horse

São os cavalos de tróia, programas que são entregues para o usuário de forma legítima (muitas vezes podem ser coisas interessantes como joguinhos, cartões virtuais, etc.), mas que internamente realizam ações maliciosas, tais como: gravar senhas, gravar toques de tecla, e posteriormente armazenar estas informações ou enviar para outra pessoa.

Vírus

São códigos ou programas que infectam outros programas e se multiplicam, na maioria das vezes podem causar danos aos sistemas infectados.

VPN:

Rede Privada Virtual – Virtual Private Network – Rede virtual criptografada criada através de um túnel numa rede IP convencional, utilizando o padrão IPSec.

Vulnerabilidade

Estado de um componente de um sistema que compromete a segurança de todo o sistema, uma vulnerabilidade existe sempre, até que seja corrigida, existem vulnerabilidades que são intrínsecas ao sistema. Um ataque explora uma vulnerabilidade.

Warez

Nome utilizado por hackers para se referir a pirataria de software.

Worm

Um worm é semelhante a um vírus, mas difere pelo fato de não necessitar de um programa específico para se infectar e reproduzir. Muitos vírus hoje, possuem a característica de worms e vice e versa.